PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
G06F 13/00

A1

(11) International Publication Number: WO 99/04344

(43) International Publication Date: 28 January 1999 (28.01.99)

(21) International Application Number: PCT/US98/14886

(22) International Filing Date: 18 July 1998 (18.07.98)

(30) Priority Data:
60/053,070 18 July 1997 (18.07.97) US

(71) Applicant *(for all designated States except US)*: NET EXCHANGE, INC. [US/US]; Suite 854, 11 Broadway, New York, NY 10004 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: MILLER, Stephen, S. [US/US]; Apartment 6K, 8 Gramercy Park South, New York, NY 10003 (US). ROSS, Lewis, Edward [US/US]; 814 Empire Avenue, Far Rockaway, NY 11691 (US). SHAALAN, Mohammed, S. [EG/US]; 800 Kearney Avenue, Kearny, NJ 07032 (US).

(74) Agent: GABLE, R., Lewis; Cowan, Liebowitz & Latman, P.C., 1133 Avenue of the Americas, New York, NY 10036-6799 (US).

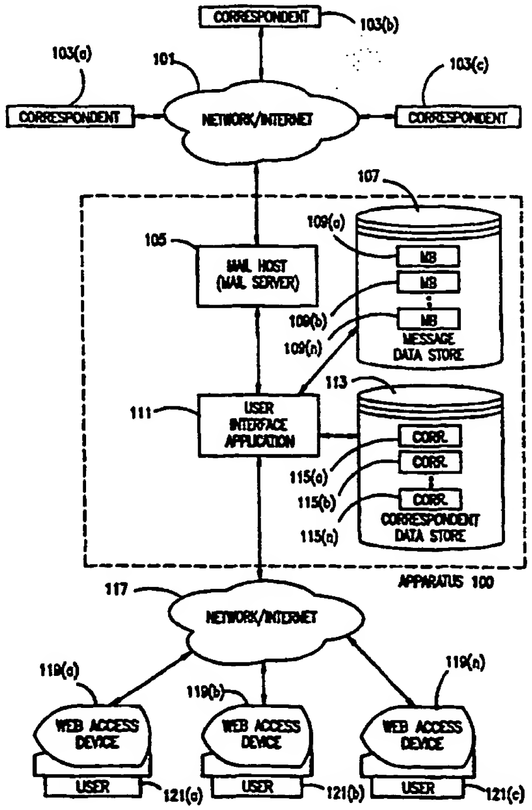
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: APPARATUS AND METHOD FOR EFFECTING CORRESPONDENT–CENTRIC ELECTRONIC MAIL

(57) Abstract

Techniques to make e-mail correspondent-centric rather than message-centric (985–999), and reduce junk e-mail (1001–1091). Tabulates (985–999), maintains (985–999), and updates (115(a), 115(b), ..., 115(n), 215) useful information about the user's chosen correspondents, and the history and status of each correspondence series. Filters incoming messages from an unrecognized sender (1013–1031, 1061–1075), asking user (1019) whether to add sender to correspondent list, and if so prompts user (1023) for needed information. Eliminates the need to search for e-mail adresses. Facilitates viewing sequential messages to and from a correspondent. Provides an effective tool to eliminate junk-mail (1013–1031, 1061–1075) by making it simpler and more practical to screen messages or change one's e-mail address. When user (121(a), 121(b), 121(c)) changes his e-mail address, automates notification of user's chosen correspondents, and in some cases can automatically update such correspondents'e-mail address lists. Eliminates need to manually create and maintain mailboxes or folders (985–999). Allows automated organization of e-mail by correspondent (701–711, 215). Is easier to learn and use than previous forms of e-mail.



APPARATUS AND METHOD FOR EFFECTING CORRESPONDENT-CENTRIC ELECTRONIC MAIL

FIELD OF THE INVENTION

This invention concerns electronic mail, and in particular a correspondent-centric way of organizing and processing e-mail to enhance setup, ease of use, convenience, storage, and functionality of e-mail. For end-users the invention simplifies and improves the management of messages and e-mail addresses, helps manage and reduce junk e-mail, and makes it easier to manage multiple mail-boxes. The invention also helps organizations set up and manage group e-mail systems with less effort and inconvenience, and at lower cost.

BACKGROUND OF THE INVENTION

E-mail is widely used today and its rapid growth is expected to continue. Over 70 million people use e-mail, sending over 200 million messages daily. Usage is expected to grow by 50% this year, with rapid growth projected for the foreseeable future

However, despite e-mail's growing popularity, current e-mail systems have various drawbacks. These include the fact that e-mail systems are hard to use (particularly for non-technical users), that users are often plagued with excessive junk e-mail, and others drawbacks which will be described below.

such as HotMail and Yahoo Mail.

The most widely used e-mail protocols today are POP3 and SMTP. POP3 ("Post Office Protocol 3", as specified in RFC 1725) is an interface standard designed to facilitate mail management locally on the user's e-mail device. Any POP3-compliant client can receive e-mail through a POP3-compliant e-mail server. (Note: a recent interface protocol, IMAP4 - RFC 1730, is similar to POP3 except that it gives the client the option of sharing additional functionality with the server.) Likewise, SMTP (Simple Mail Transfer Protocol, as specified in RFC 821) is an interface used by e-mail servers to exchange messages with other servers. In order to exchange mail over the Internet, servers in both client-server and host-based e-mail systems must be SMTP-compliant.

POP3 and SMTP-based e-mail softwares create, send, and store e-mail in a standard format that does not lend itself to certain functions (that format is specified in RFC 822). These standard e-mail messages are self-contained strings of text, delimited into several standardized fields. Key fields in the messages text string include "header" information (e.g. sender's e-mail address, recipients' e-mail addresses, date/time sent, topic, etc.), and message "body". Other fields can be appended, but are principally useful only if sender's or receiver's e-mail system can recognize and use them.

These e-mail softwares store and let the user view these correspondents, because the sender and receiver address fields allow many different text formats for messages sent to the same e-mail address.

Another problem with prior art systems is that they don't manage e-mail address lists well. Just as with handling of e-mail messages, the prior art handles e-mail address lists as flat files with no intelligent linking either to other e-mail address lists or to messages. Also, prior art e-mail address lists must be painstakingly created and managed by the user, rather than being automatically created based on correspondence.

The proliferation of junk e-mail is another problem with the prior art. Junk e-mail - often called "spam" - has lately become so pervasive that a Wall Street Journal article recently opined that spamming "has no foolproof solutions." Unfortunately, it is impossible to prevent spam by excluding messages from offending e-mailers, because spammers can easily fake their sender e-mail address. The prior art attempted to deal with spam by letting users create e-mail filters in their local e-mail system. Such a filter sorts incoming e-mail for the recipient into categories determined by the user. The filter simply scans each e-mail message as it reaches the recipient and determines what category it should be placed into. One category is, of course, "discard." Messages which the filter places in that category are automatically discarded. However, these filters have two disadvantages. First, they are hard to create, and message is stored in two to four places (e.g., in client-server systems, the message is stored on sender's client computer, recipient's client computer, and often on both sender's and receiver's server; in host-based systems, the server stores the message in a file for the sender and again in a file for the receiver). Further if a user addresses a message to ten people, then as many as 22 identical copies of that message may reside on the clients and servers of the sender and his addressees!

The second storage problem with the prior art happens when a user wants to file a message under more than one topic. The prior art does this by filing a copy of the message in each file (or folder) selected by the user. If a prior art user wants to store a message under ten topics, then ten copies of the message will be stored (and in the more recent IMAP4 systems as many as 20 copies of the message will be stored - 10 on the client and 10 on the server!).

The problems with the prior art exist because since the time of e-mail's development in the 1960's and early 1970's, e-mail has been based on the currently outdated "flat-file" database technology. Flat-file databases, also called also "non-relational" databases, store information as a simple series of "records", each containing identical "fields" of information (like subsequent rows a spreadsheet, each containing one field of information for each column of the spreadsheet). E-mail messages were structured as flat-file records - self-contained strings of text, delimited into

1. It does not organize e-mail automatically - instead requires users to organize their e-mail manually; Inboxes and Outboxes grow large and unwieldy because messages are not automatically filed;

2. Hard to see on a single screen the chronological correspondence to and from a given correspondent;

3. Users cannot view on a single screen consolidated information about their correspondence history with multiple correspondents;

4. Hard to remember or find correspondents' e-mail addresses;

5. Doesn't remind users about key information triggers, such as whether the last correspondence with a party was incoming or outgoing, and which correspondences have lapsed.

6. Hard to find past messages;

7. Hard to view groups of past messages in meaningful ways;

8. Users can view messages from only one folder at a time;

9. Time consuming to set up, maintain, and use multiple e-mail address lists;

10. Hard to identify or screen junk e-mail;

interface which has made it easier to write e-mail programs, but not easier to use them. Problems with prior art e-mail systems include the following: they are hard to use, don't manage messages in optimal ways, fail to manage e-mail addresses well, suffer from excess junk e-mail, make it difficult to manage multiple mailboxes, and are inconvenient for organizations to set up and maintain.

OBJECTS AND ADVANTAGES

The object of the invention is to provide a simple, easy-to-use, intuitive e-mail system with enhanced protections from junk e-mail, and which overcomes various drawbacks of prior art e-mail systems.

Accordingly, several objects of the invention are as follows:

1. View consolidated information about their correspondence history with all correspondents.
2. Easily view a chronological correspondence to and from a given correspondent.
3. Avoid the inconvenience of remembering or looking up e-mail addresses.
4. Eliminate or reduce junk e-mail by either screening incoming mail by correspondent, or conveniently changing one's e-mail address while simultaneously effecting the change in the systems of desired correspondents.
5. Have their e-mail organized automatically by the system, rather than having to organize it manually.

FIG. 8 is a user screen showing the "Change E-mail Address" option for the preferred embodiment.

FIG. 9A is a high level system architecture diagram of the invention.

FIG. 9B is a functional block diagram of the internal structure of the incoming message server.

FIG. 9C is a functional block diagram of the queue manager server.

FIG. 9D is a functional block diagram of the internal structure of the mass storage server.

FIG. 9E is a functional block diagram illustrating the outgoing queue manager/message server.

FIGS. 9F and G illustrate respectively a generalized and a more particular diagram of the data tables comprising the mass storage and the relationships between the data tables.

FIG. 9H is an object relation diagram which illustrates the structure of the message object.

FIGS. 10A and B comprise together a flow diagram showing how an input message is processed.

FIG. 10C is a flow diagram showing how an output message to be transmitted is processed.

FIGS. 11A, B, C and D respectively illustrate the data structure of a request for retrieving a message, a correspondent information request, a correspondent message history request and a topic content request.

FIG. 12 is a table showing all of the correspondent

This embodiment assumes that clients will access their e-mail through the Internet using a Web browser installed on any Internet access device. (This configuration will be further described below.)

Apparatus **100** is employed in network **101** which connects any number of e-mail users or correspondents **103 (a ... n)**. Network **101** may be the Internet, a commercial e-mail network, or a privately owned network system. Each correspondent **103** is connected to network **101** by means of a link over which the correspondent **103** can send and receive e-mail messages. Mail or message items are sent by correspondents **103** to and from each other. Apparatus **100** allows users **121 (a ... n)** to send and receive e-mail messages of whatever type used in the network (typically internet mail standard messages).

When a new message is received by apparatus **100** from network **101**, it is intercepted by mail host **105** (also called a mail server). Mail host **105** can be any computer configured as a mail server or mail host, having e-mail server software installed, such as Sendmail (for UNIX servers), other internet standard mail servers, or a proprietary mail server such as Lotus Notes, CC Mail, or Microsoft Exchange. When mail host **105** receives an incoming message from Network **101** it handles the message in the standard way, identifying the appropriate recipient. However, traditionally mail host **105** would post the message directly to the message data store could be a personal computers, network computers, televisions with WebTV units attached, Web telephones, or other Web access devices which are currently being developed.

When an e-mail user uses apparatus **100** to access his e-mail, he will use his Web browser to link to apparatus **100** through Network/Internet **117**. When he links to user interface application **111** he will see, using his web browser software, an interface which combines information from message data store **107** with correspondent data store **113**. This combination allows novel views of e-mail such as those shown in **FIGS. 5 - 8**.

When an e-mail user uses apparatus **100** to send an e-mail message, the message is posted to message data store **107**, and in addition, information from the message is used to update correspondent data store **113**. The message is then sent to the appropriate recipient through either network **101** or network **117**, as appropriate.

FIG. 2 shows a high-level overview of another embodiment of the invention, shown as apparatus **200**. This embodiment assumes that e-mail users will have an e-mail software which embodies the invention installed on their local computer (more about this below).

Most of the components of **FIG. 2** are similar to those of **FIG. 1**, and are labeled with the same numbers except that the users **121** **(a...n)** or user **221** corresponds. Each subsequent line in the table describes the specific information for each of correspondents **103** **(a...n)** or **203** **(a...n)**.

FIG. 4 shows new message processing **400**. For incoming messages, new message processing **400** is applied to each message to assure that, before saving the message to message store **109** **(a...n)** or **209**, the message is linked to the appropriate correspondence record in correspondent data store **115** **(a...n)** or **215**, and so that the correspondent data store record can be updated.

New message processing **400** starts after a message is received by mail host **105** or **205** and has been transmitted by the mail host to user interface application **111** or **211**. We will assume here that the new incoming message is addressed to user **121(a)** or **221(a)**. Such message would have either been sent through network **101** or **201** from a correspondent **103** **(a...n)** or **203** **(a...n)**, or alternatively from a user **121** **(b...n)** or **221** **(b...n)**, transmitted through network **117** or **217**. Upon receipt of this message, mail host **105** or **205** would transmit the message to user interface application **111** (in the case of apparatus **100**) or through network **217** to user interface application **211** (in the case of apparatus **200**). Upon receipt, user interface application **111** or **211** would begin new message processing **400**.

to determine if there is information to guess the name of the sender. For example, the name of the sender is often included within <...> brackets in the sender e-mail address field. If the answer to step **407** is yes, this information is temporarily stored as default sender name. Otherwise, step **409** is applied to temporarily store a generic sender name (such as "unrecognized sender," or "?") as the default sender name for the message.

Step **411** then prompts the user whether he/she wants to store or delete the message. (In making this decision the user can optionally read the text of the message.) If the user response in step **413** is "delete," step **415** deletes the message. If the user's response in step **413** is "store", user interface application **111/211** proceeds to process step **417**.

Step **417** displays the currently stored default sender name for the message in a text box which can be revised by the user. Step **417** also asks the user to perform step **419**, in which the user either accepts the default sender name, or revises it and confirm the revision.

The user interface application **111/211** then performs step **421**, which is to save the message to the message data store 107, noting the record number of the newly saved record, which will be used in step **423**.

In step **423** a new record is created in the correspondent

correspondent table in **FIG. 5**. Option **505** allows the sender to automatically delete these two messages from unrecognized senders.

FIG. 6 shows the user screen seen when the user clicked on the first line in table **500**, line **501**. Note that the user sees not only the message from the sender indicated in **501**, but he also sees past incoming and outgoing correspondence, in reverse chronological order, with that sender.

FIG. 7 shows the screen the user sees when he clicks on line **507** in **FIG. 5**. The user can instantly open a pre-addressed e-mail screen to communicate with any user in column **701** by clicking on the user's name. The user can open an e-mail window pre-addressed to multiple users by clicking on boxes in the three columns in **703**, then clicking on the confirm button **711** below. Note also that the user can see the date of his last incoming or outgoing message with each correspondent by looking in column **705**. Further, the user can see whether that message was incoming or outgoing by looking in column **707**. And the user can also see how many previous incoming or outgoing messages are on file for each correspondent by looking in column **709**. Each of these capabilities are made possible by referencing the information in the table in **FIG. 3**, reflecting correspondent data stores **115** for the respective user, or **215**.

FIG. 8 shows a user screen which can be used to decisions in a fraction of the time required to re-scan the message every time searching for a field.

After a Message has been converted to a Message Object, the incoming Message Server (903) sends the object to one of one (or more) identical Queue Manager servers (907). The function of the Queue Manager is to sort messages according to a given priority algorithm, then send them one at a time to the Mass Storage Server (909). If one Queue Manager server becomes overloaded, some of the objects on this server will migrate to another Queue Manager server according to a given algorithm.

Mass Storage is where all data and system information is stored, searched, and updated through Queue Manager servers (907) and Application Servers (913)

An Application Server is responsible for providing transformations upon Message Objects moving between User Interface Servers (915) on one hand, and the Mass Storage Server (909) and Outgoing Queue Manager servers (917) on the other hand. Also, the Application Server communicates with a State Server (913) to temporarily store current login information about a specific user. The State Servers (913) and Application Servers (911) together provide a way of keeping track of user activity or state during a given session. The State is stored for a limited amount of time before being discarded.

firewall, thus providing a high level of security for data stored on the Mass Storage (909).

Figure 9-B shows the internal structure of the Incoming Message Server (903). As shown in figure 9-B, the Incoming Message (901) is delivered to a Device-Specific Driver/Daemon (931) which handles transport media-dependent incoming messages according to their media (e-mail SMTP daemon, Fax Receiver, etc.).

After being converted to a stream, file, or other standard input forms, the Message is passed to a Local Delivery Agent (933), which receives a request from a Device-Specific Driver (931) to deliver a message to the local machine users. A local delivery agent converts the message from media-dependent to a stream format, and sends that to a Message Parser (935).

The Message parser 935 converts the message stream to a media-independent message object.

Through parsing, the message key fields are extracted from message headers and stored in message object properties (attributes) to be accessed by other system components. After the message object has been populated with data, it is then sent to an Object Trading Layer (937) which is responsible for delivering a given message object to the least loaded Queue Manager server (907) according to work load statistics provided by the Queue Managers (907) sends a series of remote method invocations to the Mass Storage Interface (951), which in turn knows how to deal with the internal structure of the Mass Storage.

Figure 9D shows the internal structure of the Mass Storage Server (909). As shown in figure 9D, a Mass Storage Interface 951 provides high level methods that will be called by the Object Insertion Module (945) through RMI (Remote Method Invocation) to store Message Objects. The Mass Storage Interface 951 is the responsible for the actual communication with the Mass Storage Server, also referred to as the Database (953). The Mass Storage (953) is the actual location for storing and manipulating users' Messages, Correspondents, and Topic information. See figure 9-F for details on the entity relationship diagram of the database.

Figure 9E shows the internal structure of the Outgoing Queue Manager/Message Server (917).

As shown in figure 9E, an Object Trading/Migration Layer (961) communicates with the Object Trading Layer of the Application Server (911). Both layers work to deliver Message Objects with embedded outgoing message information.

The Migration Layer communicates with the Object Queue Manager (963) to deliver objects to the next unloaded Queue Manager. The Object Trading/Migration Layer (961) passes the message to an Outgoing queue Manager (963) which holds

Figure 9-G is lower level entity-relationship diagram. The blocks shown in 9G represent the same data tables as those in Figure 9-F, with the only difference being that in 9G each block contains additional description about the information stored within the data table represented by that block.

In both 9F and 9G, lines 999 connecting pairs of data tables indicate that those two tables are "related," which means that the records in one data table may be linked to records in the other. The connecting points of these lines sometimes fork into three prongs, which indicates that multiple records from a table so marked may be linked to a single record in the related table - a "many-to-one" relationship. the relationship is also indicated by the digits "m" (many) or "1" (one) next to the point where each relationship line intersects with a block representing a table.

The block or data table numbers in 9F and 9G are identical, except that the data tables in 9G are labeled with a "'" symbol (e.g., data table 985' in Figure 9F is labeled 985' (with an apostrophe) in 9G.

As shown in figure 9-G, User table (985) maintains information about each user of the invention. User ID is a unique identifier for that user. Other information in the address. Other information about correspondents in the User-Correspondent data table may include first and last name, description, comments, phone, address, etc.

Note that Correspondent data table 989 embodies several key innovations in the Invention. (1) Whereas in the prior art, each e-mail address on an e-mail address list must be consciously entered by the user, in the Invention the Correspondent data table becomes an e-mail address list, and the system automatically creates posts an entry to the Correspondent data table for any message sent to or accepted from a correspondent not already contained in the Correspondent data table (see more about this process in Figures 10A, 10B, and 10C below). This feature greatly simplifies the task of keeping track of e-mail addresses. (2) Correspondent data table 989 can maintain additional information about correspondents, which can be displayed in helpful ways. For example,while prior art messages ofter come from a sender whose identity is not readily recognizable, a User of the Invention can identify a name or descriptive term for each correspondent, so that upon receiving a message from something like as jxam5@domain.com, the system will inform the User that the message is from RealName@domain.com. (3) The Correspondent data table gives the Invention a completely new and powerful way to identify and deal with junk e-mail. Where as all junk-mail filtering systems to date are "negative filters" (i.e., they search for information within a message to be used to identify the pointer information to a single message in Message data table 993 and a single correspondent in Correspondent data table 989.

Note that the Message-Correspondent Relationship data table 995 is a key innovation in the Invention. Whereas in prior art e-mail systems at least one instance of a message must be stored on a computer somewhere for every party to a message (i.e. the sender and each address), in the Invention the message is stored only once, without regard to the number of parties to the message. The Invention accomplishes this result by replacing the prior art's multiple instances of the same message, with a single copy of the message, and multiple instances of only short pointer records, which are stored in the Message-Correspondent Relationship data table 995.

Another key innovation of the Invention which is embodied in the Message - Correspondent Relationship data table 995 is that the Invention can automatically link all messages to and from a given correspondent. This facilitates unique reports such as Figures 6, as well as columns 705, 707, and 709 in Figure 7.

Topic data table 991 represents topics which users can create to categorize their messages, so that it is easier to retrieve messages when they are needed in the future. This table contains a list of all topics which a User has created for incoming and outing messages pertaining to to Email Box

Figure 9-H is an object relation diagram which describes the structure of the Message Object. The Message Object represents the information contained in the message string, however in a more readable format. Using this format, it is easier for the system to handle logic decisions in a fraction of the time required to re-scan the message every time searching for a field.

As shown in figure 9-H, the Message Object (1101) is composed of a set of properties, and two vectors (dynamic arrays). The first vector is the recipient vector (1103) which contains a number of recipient objects (1105). The second vector is the attachment vector (1107) which contains a number of attachments object (1109).

The Properties of Message Object (1101) includes.

CharSet	Character set used to compose the
MessageHeader	Contain complete header of the message
SubType	Content Subtype
Type	Content Type
Date	Date Message was sent
FromFirstName	Senders First Name
FromLastName	Senders Last Name
FromAddress	Senders e-mail address
MessageID	Message Universal unique ID
MessageBody	Contain body of message
Priority	Message priority
ReplyToName	Reply to name
ReplyToAddress	Reply to E-mail address
Subject	Message Subject
MimeVersion	Contain information about MIME format

The properties of Recipient Object (1105) includes.

used for web surfing or in correspondence with un-trusted correspondents), or the e-mail is a trusted e-mail address (i.e. used in correspondence with trusted correspondents).

In case of an un-trusted e-mail address, the database is searched in step 1013 for a matching correspondent address in the correspondent data store (989 of figure 9-G) if the correspondent address exists, then the message is saved in step 1015 into the message data store 993 of figure 9-G, the relation between the user, correspondent and the message is stored by step 1017 in the Message - Correspondent data store 995 of figure 9-G.

However if the correspondent address does not exist in the correspondent data store, then step 1019 prompts the user to either store or delete the message. If the user response was to delete the message, then step 1031 deletes the message from the message data store.

If the user response is to store the message, step 1023 prompts the user to either accept or revise the user information before being stored in the correspondent data store. Step 1029 stores the message in the message data store and the relation between the user, correspondent and the message is stored by step 1029 in the Message - Correspondent data store.

Trusted e-mail is processed as shown in Figure 10-B.

2- The user chooses by step 1077 to send the message back to the correspondent with a message stating that user does not exist on the server (bounce the message back).

3- The user chooses by step 1079 to forward the message to another e-mail address.

Figure 10C explains the business logic applied to an outgoing message. When user sends a message to the system as shown in figure 10C, the message is parsed in step 1083 and a Message Object is created. The Message Object represents the information contained in the message string, however in a more readable format. Using this format, it is easier to the system to handle logic decisions in a fraction of the time required to re-scan the message every time searching for a field.

Having the message object populated with key fields, step 1087 makes a search in the database to determine if the correspondent e-mail address exists in the Correspondent data store. If that address exists, then the message is saved by step 1089 to the message data store, the relation between the user, the correspondent and the message is stored in step 1091 in the Message - Correspondent data store.

If the recipient address was not found in the correspondent data store, then the user is requested by step 1093 to either accept or revise recipient information before being stored by step 1095 in the correspondent data store.

e-mail address id and topic id . Also 2 counter fields are presented, count 1 is used to indicate the number of messages needs to be displayed in detailed format, count 2 is used to indicate the number of messages needed to be displayed in summery format. Every two fields are separated by a separation indicator (-).

CONCLUSION

The invention operates by taking a novel approach to e-mail from the approach in use today. Current e-mail systems, including the user interfaces they provide, take a message-centric approach to e-mail - e-mail is sorted, stored, and shown in an exclusively message-centered way, with no attention to helping the user keep track of correspondent-centered information. The invention lets the user add and maintain correspondent-centered information to the e-mail system, and take advantage of the various user-interface and privacy benefits that this approach offers.

in the second memory and, if there is a match, storing the second portion of that matching message in the second memory.

3. A method of receiving and evaluating the transmitted message as claimed in claim 2, where in step g) if there is no match, displaying an indication of the transmitted message to prompt the message user to accept or reject the transmitted message.

4. A method of evaluating and transmitting a message from a message user to a selected correspondent, said message comprising a first portion comprising the message content and at least one further second portion comprising the address of the correspondent for which the message is intended, said method comprising the steps of:

a) storing in a second memory the addresses of the correspondents to which
messages have been previously transmitted;

b) isolating the identified first portion from the second portion of the message to be transmitted;

c) evaluating the message to be transmitted to identify the second portion;

d) comparing the correspondent address of the message to be transmitted with the correspondent addresses stored in the second memory;

e) if there is a match of correspondent addresses in step d), storing a single instance of said first

FIG. 1

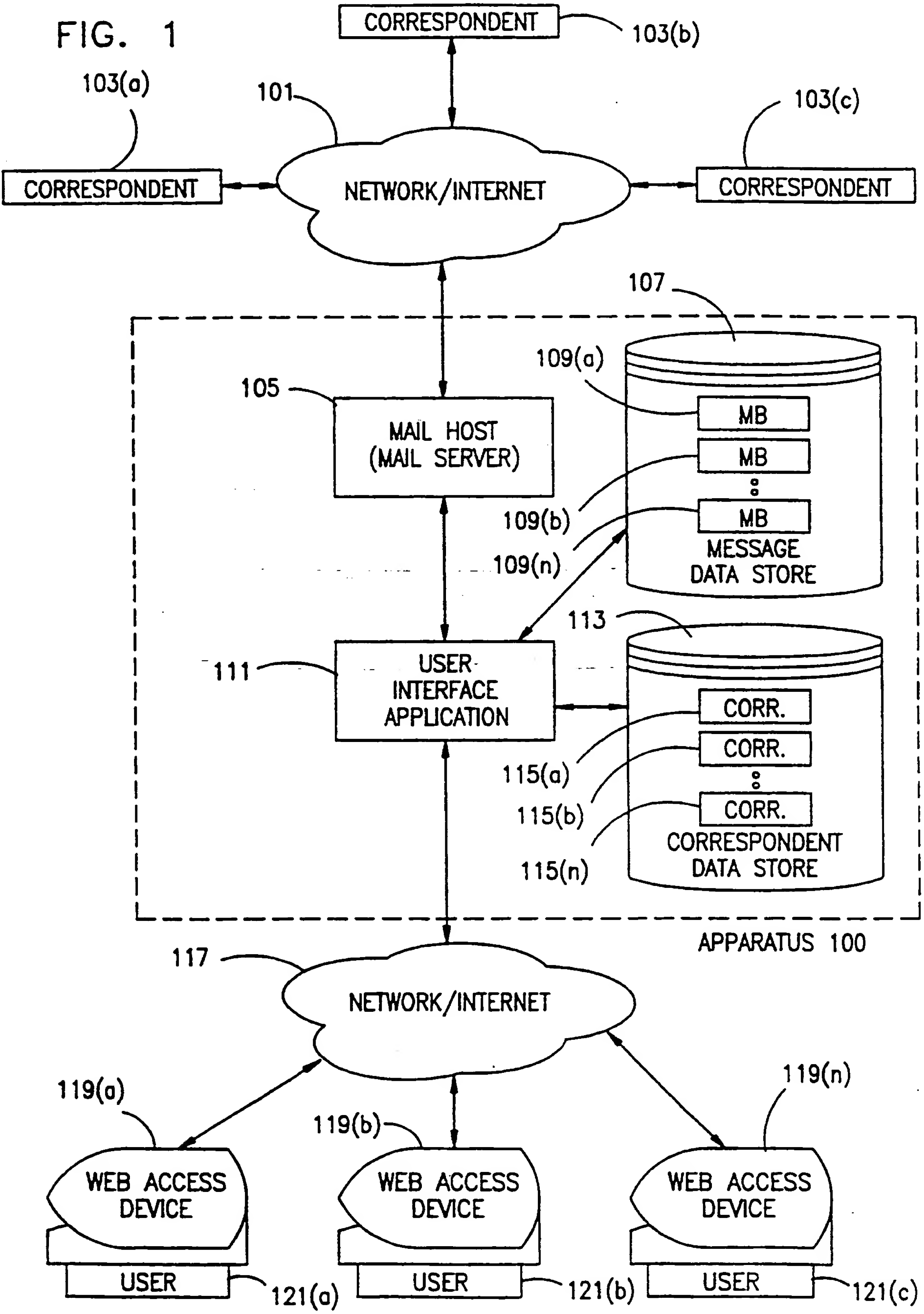
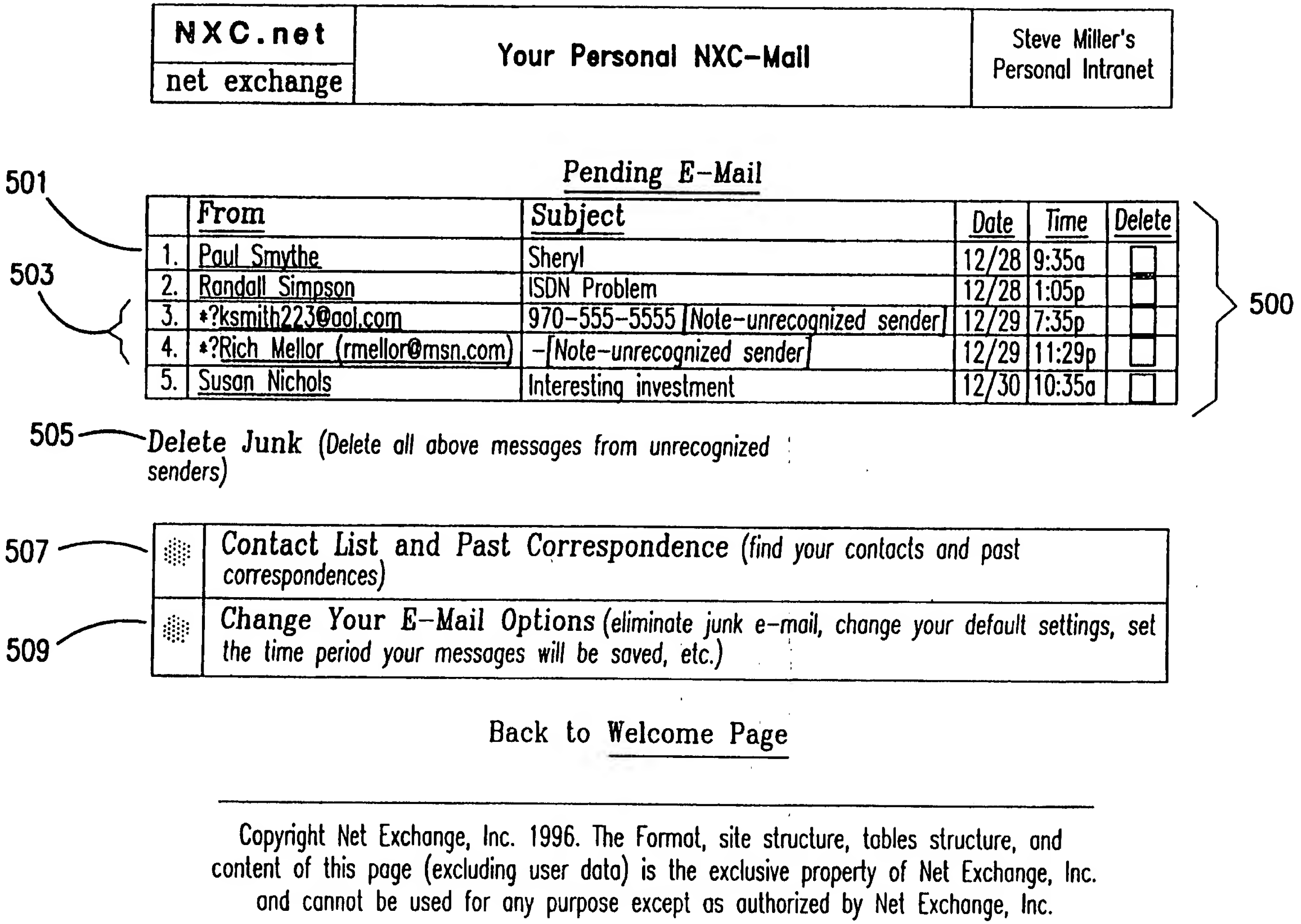

CORRESPONDENT
103(b)
103(a)
101
103(c)
CORRESPONDENT
NETWORK/INTERNET
CORRESPONDENT
107
109(a)
105
MB
MAIL HOST
(MAIL SERVER)
MB
109(b)
MB
109(n)
MESSAGE
DATA STORE
113
111
USER
INTERFACE
APPLICATION
CORR.
CORR.
115(a)
115(b)
CORR.
CORRESPONDENT
DATA STORE
115(n)
APPARATUS 100
117
NETWORK/INTERNET
119(a)
119(n)
119(b)
WEB ACCESS
DEVICE
WEB ACCESS
DEVICE
WEB ACCESS
DEVICE
USER
121(a)
USER
121(b)
USER
121(c)

FIG. 3

CORRESPONDENT TABLE (CORRESPONDENT DATA STORE)

II5 (n) or 215

CORRESPONDENT NAME	E-MAIL ADDRESS	LINKS TO MSGS. IN MSG. DATABASE	# OF MESSAGES IN DATABASE	LAST MESSAGE TYPE (IN OR OUT)	DATE OF LAST CORRESPONDENCE
CORRESPONDENT 1	ADDRESS FOR CORR. 1	MSG. ID'S FOR CORR. 1	# OF MSG.'S TO & FROM CORR.1	LAST MSG. STATUS, CORR. 1	DATE FOR CORR. 1
CORRESPONDENT 2	ADDRESS FOR CORR. 2	MSG. ID'S FOR CORR. 2	# OF MSG.'S TO & FROM CORR.2	LAST MSG. STATUS, CORR. 2	DATE FOR CORR. 2
...	...	...	...	...	...
CORRESPONDENT n	ADDRESS FOR CORR. n	MSG. ID'S FOR CORR. n	# OF MSG.'S TO & FROM CORR. n	LAST MSG. STATUS, CORR. n	DATE FOR CORR. n



NXC.net net exchange	Your Personal NXC-Mail	Steve Miller's Personal Intranet

Pending E-Mail

	From	Subject	Date	Time	Delete
1.	Paul Smythe	Sheryl	12/28	9:35a	□
2.	Randall Simpson	ISDN Problem	12/28	1:05p	□
3.	*?ksmith223@aol.com	970-555-5555 [Note-unrecognized sender]	12/29	7:35p	□
4.	*?Rich Mellor (rmellor@msn.com)	-[Note-unrecognized sender]	12/29	11:29p	□
5.	Susan Nichols	Interesting investment	12/30	10:35a	□

Delete Junk (Delete all above messages from unrecognized senders)

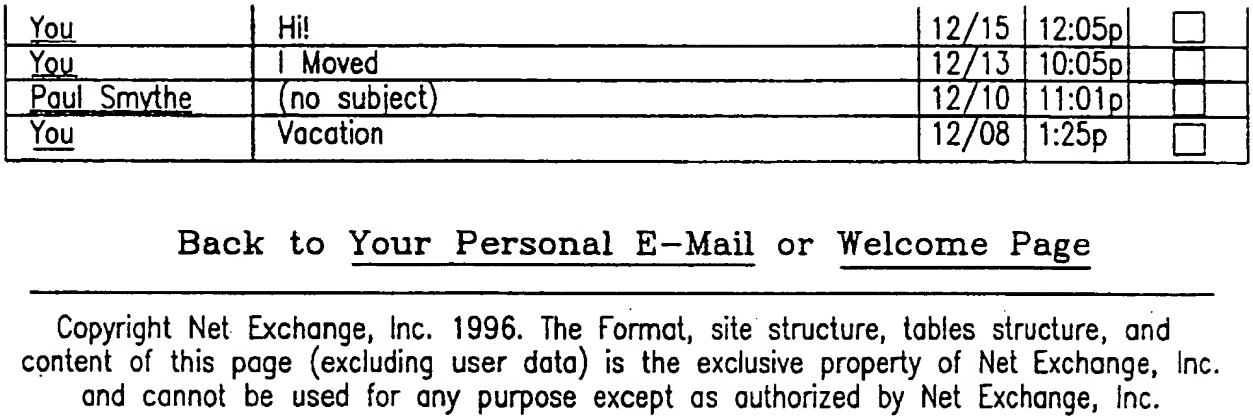
▨	Contact List and Past Correspondence (find your contacts and past correspondences)
▨	Change Your E-Mail Options (eliminate junk e-mail, change your default settings, set the time period your messages will be saved, etc.)

Back to Welcome Page

Copyright Net Exchange, Inc. 1996. The Format, site structure, tables structure, and content of this page (excluding user data) is the exclusive property of Net Exchange, Inc. and cannot be used for any purpose except as authorized by Net Exchange, Inc.

FIG. 5

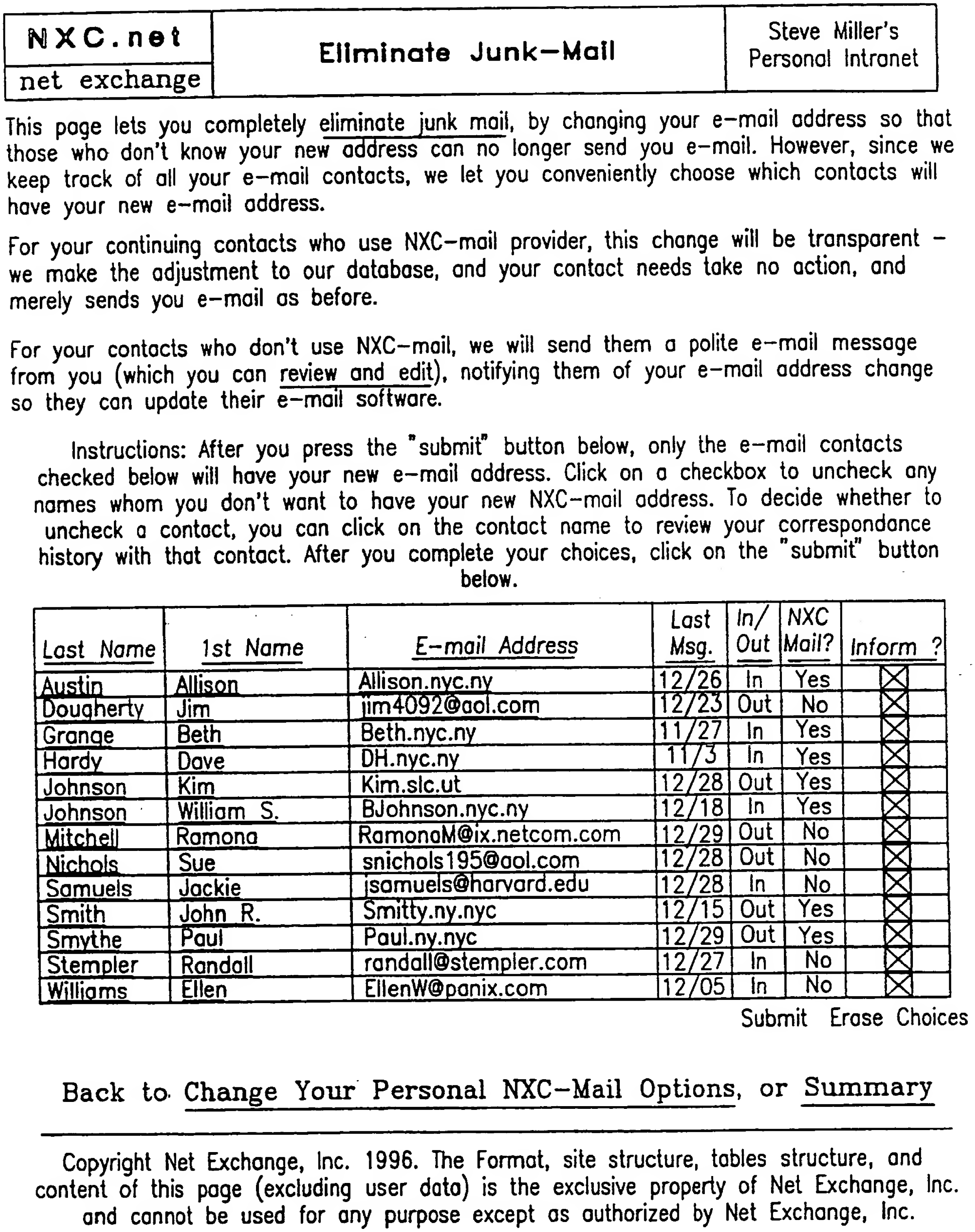
FIG. 6(b)



You	Hi!	12/15	12:05p	□
You	I Moved	12/13	10:05p	□
Paul Smythe	(no subject)	12/10	11:01p	□
You	Vacation	12/08	1:25p	□

Back to Your Personal E-Mail or Welcome Page

Copyright Net Exchange, Inc. 1996. The Format, site structure, tables structure, and content of this page (excluding user data) is the exclusive property of Net Exchange, Inc. and cannot be used for any purpose except as authorized by Net Exchange, Inc.



NXC.net net exchange	Eliminate Junk-Mail	Steve Miller's Personal Intranet

This page lets you completely eliminate junk mail, by changing your e-mail address so that those who don't know your new address can no longer send you e-mail. However, since we keep track of all your e-mail contacts, we let you conveniently choose which contacts will have your new e-mail address.

For your continuing contacts who use NXC-mail provider, this change will be transparent - we make the adjustment to our database, and your contact needs take no action, and merely sends you e-mail as before.

For your contacts who don't use NXC-mail, we will send them a polite e-mail message from you (which you can review and edit), notifying them of your e-mail address change so they can update their e-mail software.

Instructions: After you press the "submit" button below, only the e-mail contacts checked below will have your new e-mail address. Click on a checkbox to uncheck any names whom you don't want to have your new NXC-mail address. To decide whether to uncheck a contact, you can click on the contact name to review your correspondance history with that contact. After you complete your choices, click on the "submit" button below.

Last Name	1st Name	E-mail Address	Last Msg.	In/ Out	NXC Mail?	Inform ?
Austin	Allison	Allison.nyc.ny	12/26	In	Yes	☒
Dougherty	Jim	jim4092@aol.com	12/23	Out	No	☒
Grange	Beth	Beth.nyc.ny	11/27	In	Yes	☒
Hardy	Dave	DH.nyc.ny	11/3	In	Yes	☒
Johnson	Kim	Kim.slc.ut	12/28	Out	Yes	☒
Johnson	William S.	BJohnson.nyc.ny	12/18	In	Yes	☒
Mitchell	Ramona	RamonaM@ix.netcom.com	12/29	Out	No	☒
Nichols	Sue	snichols195@aol.com	12/28	Out	No	☒
Samuels	Jackie	jsamuels@harvard.edu	12/28	In	No	☒
Smith	John R.	Smitty.ny.nyc	12/15	Out	Yes	☒
Smythe	Paul	Paul.ny.nyc	12/29	Out	Yes	☒
Stempler	Randall	randall@stempler.com	12/27	In	No	☒
Williams	Ellen	EllenW@panix.com	12/05	In	No	☒

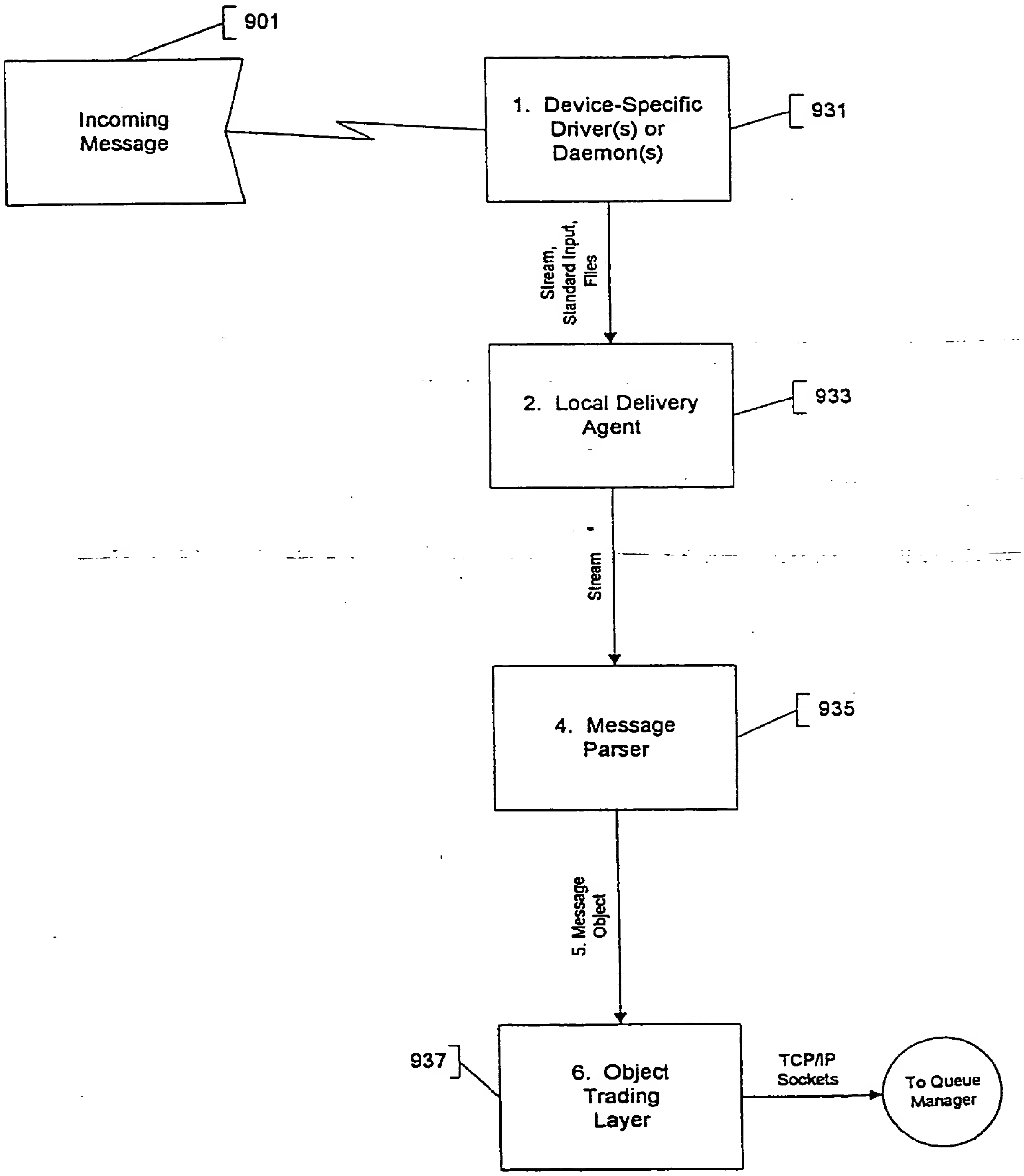
Submit Erase Choices

Back to Change Your Personal NXC-Mail Options, or Summary

Copyright Net Exchange, Inc. 1996. The Format, site structure, tables structure, and content of this page (excluding user data) is the exclusive property of Net Exchange, Inc. and cannot be used for any purpose except as authorized by Net Exchange, Inc.

FIG. 8

Figure 9 - B


901
Incoming Message
1. Device-Specific Driver(s) or Daemon(s)
931
Stream, Standard Input, Files
2. Local Delivery Agent
933
Stream
4. Message Parser
935
5. Message Object
937
6. Object Trading Layer
TCP/IP Sockets
To Queue Manager

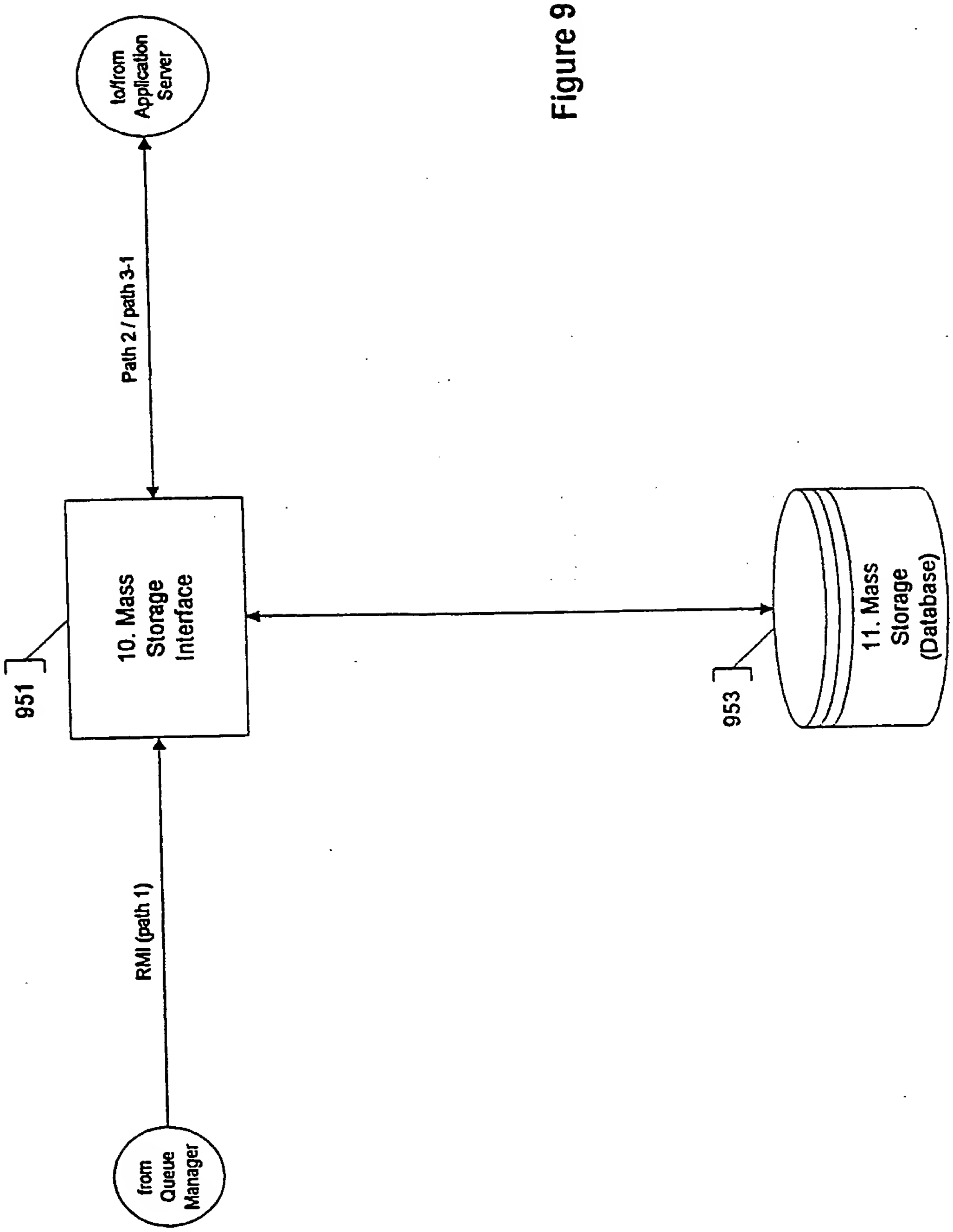

from Queue Manager
RMI (path 1)
951
10. Mass Storage Interface
Path 2 / path 3-1
to/from Application Server
953
11. Mass Storage (Database)

Figure 9 - D

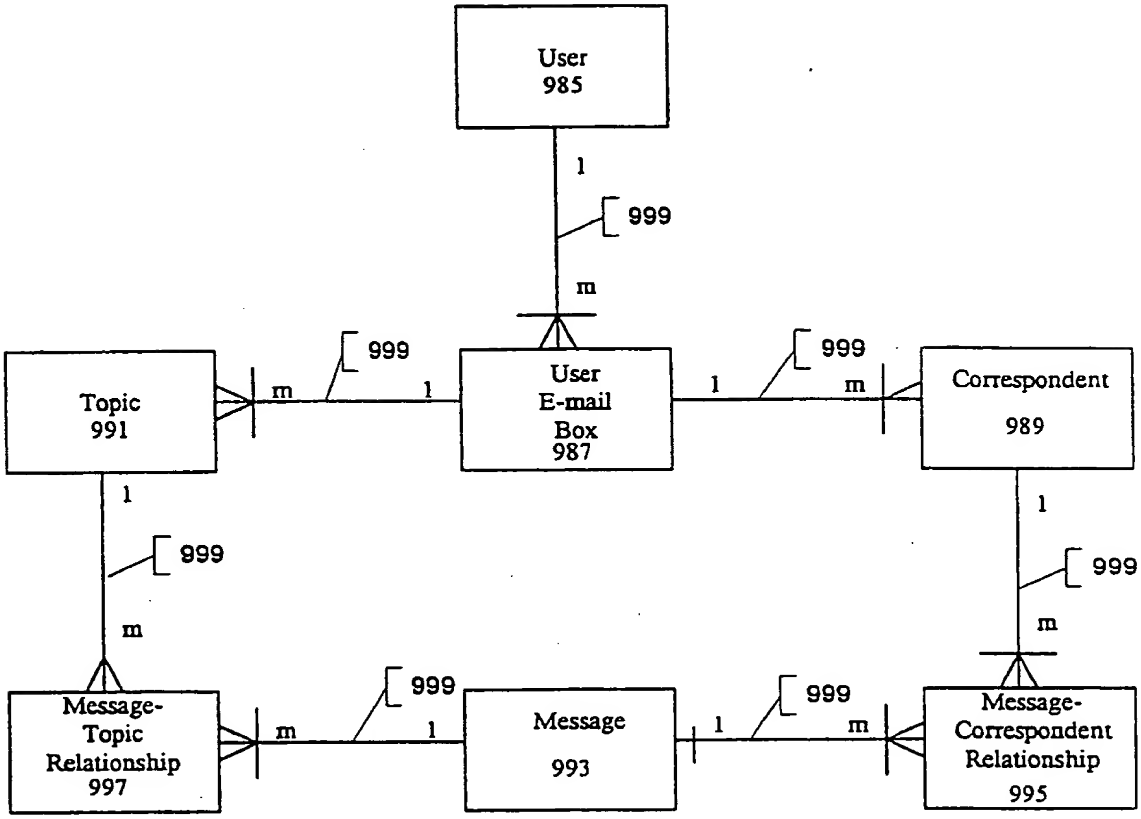

User
985
1
999
m
Topic
991
m
999
1
User
E-mail
Box
987
1
999
m
Correspondent
989
1
999
m
1
999
m
Message-
Topic
Relationship
997
m
999
1
Message
993
1
999
m
Message-
Correspondent
Relationship
995

Figure 9 - F

Figure 9 - H

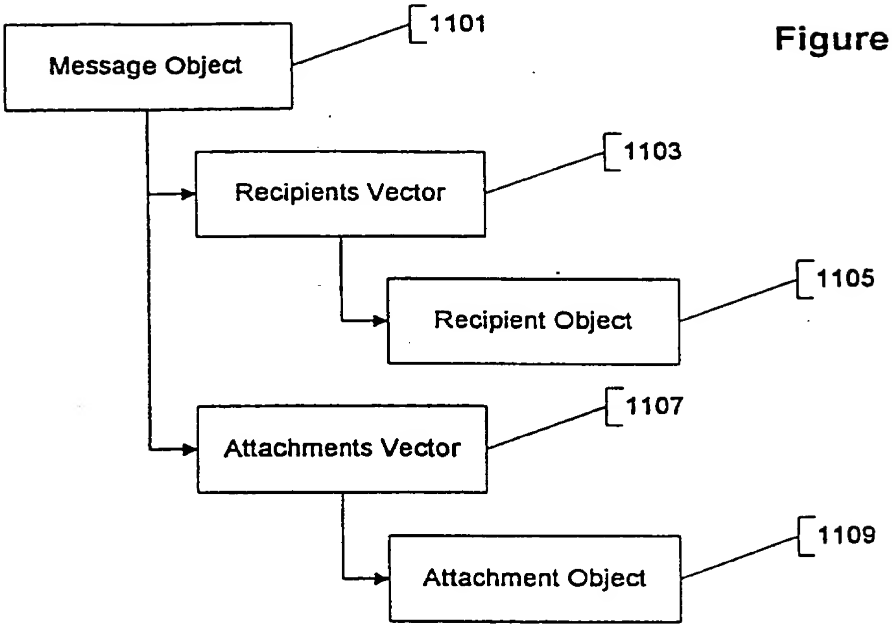

Message Object
1101
Recipients Vector
1103
Recipient Object
1105
Attachments Vector
1107
Attachment Object
1109

Figure 10 - B

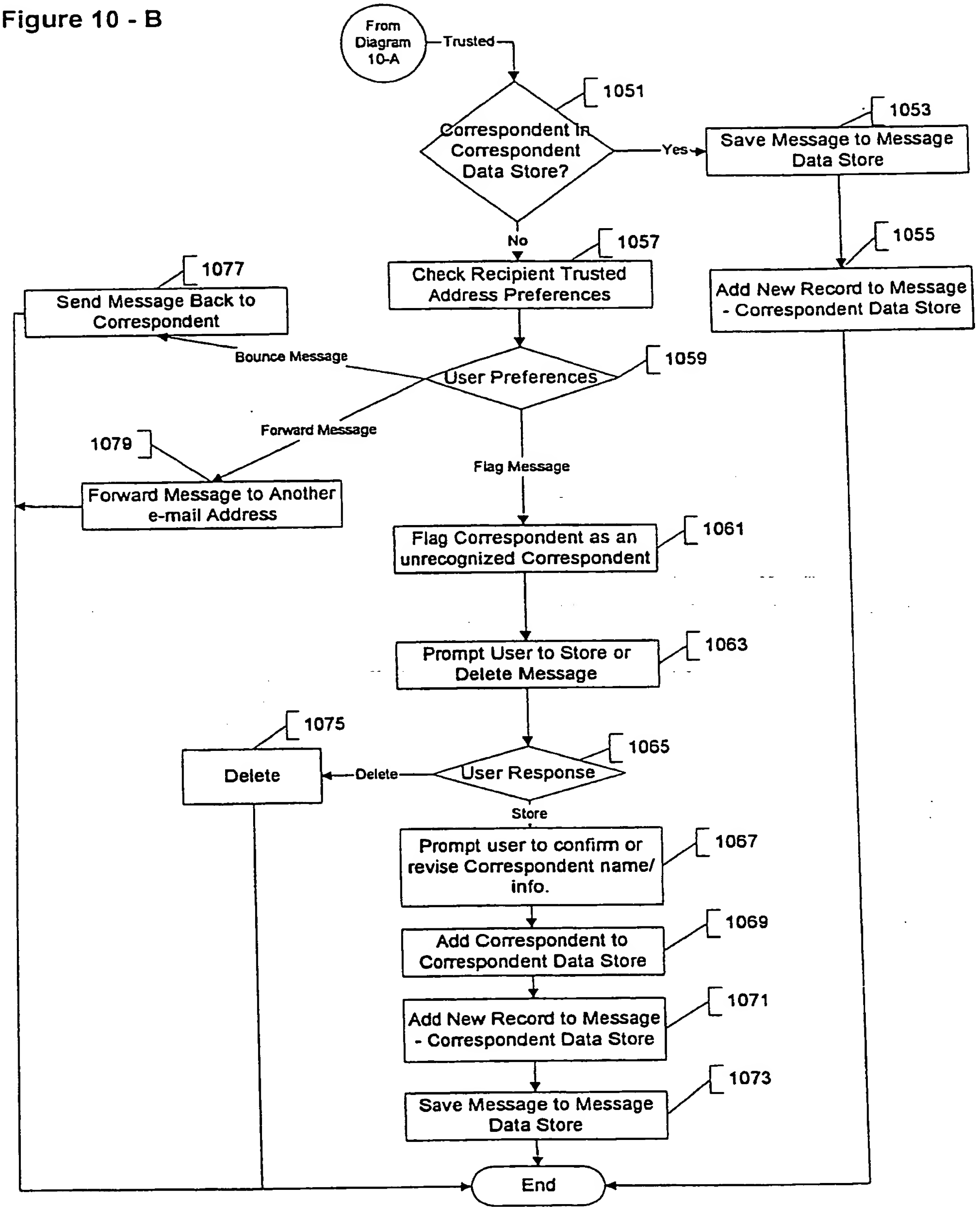

From Diagram 10-A
Trusted
Correspondent in Correspondent Data Store?
1051
Yes
Save Message to Message Data Store
1053
Add New Record to Message - Correspondent Data Store
1055
No
Check Recipient Trusted Address Preferences
1057
Send Message Back to Correspondent
1077
Bounce Message
User Preferences
1059
Forward Message
1079
Forward Message to Another e-mail Address
Flag Message
Flag Correspondent as an unrecognized Correspondent
1061
Prompt User to Store or Delete Message
1063
Delete
1075
Delete
User Response
1065
Store
Prompt user to confirm or revise Correspondent name/ info.
1067
Add Correspondent to Correspondent Data Store
1069
Add New Record to Message - Correspondent Data Store
1071
Save Message to Message Data Store
1073
End

Figure 11 A

Message Request

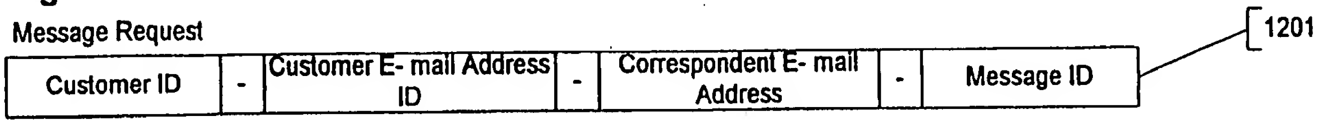


Figure 11 B

Correspondent Information Request

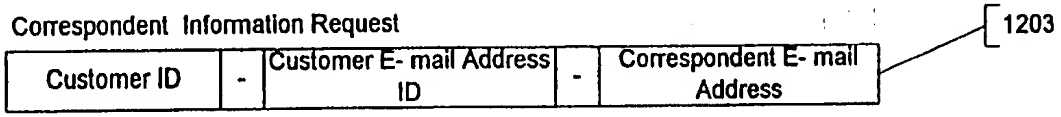


Figure 11 C

Correspondent History Request

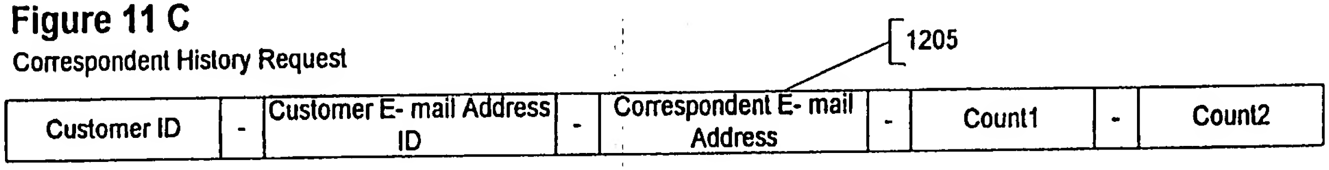
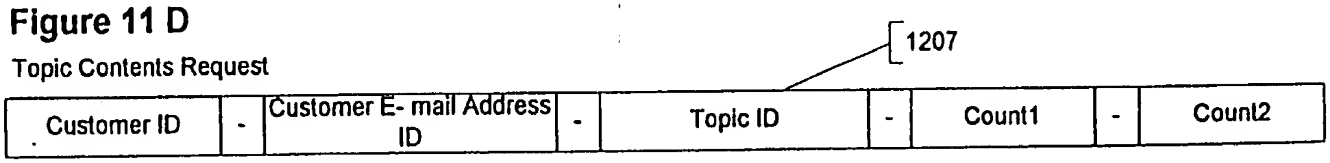


Figure 11 D

Topic Contents Request


Customer ID
-
Customer E- mail Address ID
-
Topic ID
-
Count1
-
Count2
1207